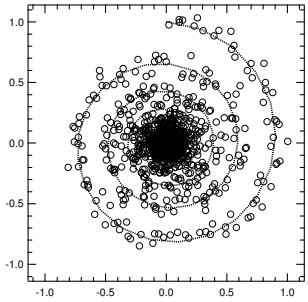
École normale supérieure
Centre de préparation à l'Agrégation de Physique

IGOR Pro Version 4 : Prise en main & Acquisition de données



Jean Hare

15 septembre 2005

Introduction

Le premier objectif de ce document est de présenter en quelques pages les notions indispensables pour une prise en main aisée du programme IGOR Pro (version 4) de la société WaveMetrics[1]. En second lieu, il constitue une introduction générale à l'acquisition de données par ordinateur, déboucant sur une desciption sommaire des programmes d'acquisition développés sous IGOR Pro que nous mettons à votre disposition. Pour leur utilisation, on se reportera à la notice détaillée de chacun d'entre eux.

A – PRISE EN MAIN

Le logiciel IGOR Pro (fonctionnant aussi bien sous Windows que sous MacOS) se présente en premier lieu comme un « grapheur » scientifique à interface graphique, parmi beaucoup d'autres. Il permet ainsi de nombreuses représentations graphiques de toutes sortes de données, et leur analyse par des outils de traitement numérique, comme les ajustements de courbes, des intégrations ou dérivations numériques, lissage, convolution et déconvolution, transformation de Fourier discrète, tests statistiques, etc...

Il s'en distingue toutefois par plusieurs caractéristiques essentielles :

- Il permet de manipuler des données en **quantités** très importantes, limitées seulement par la mémoire de la machine, et cela avec une **rapidité** inégalée. En particulier il peut acquérir et traiter des **images**, et des jeux de données de plus grande dimensionnalité, comme des enregistrements vidéo par exemple.
- En plus des méthodes « graphiques », il peut être **programmé** à l'aide d'un langage de script en ligne de commande, ce qui permet de réaliser très simplement et très rapidement des « macrocommandes », pour le traitement de jeux de données successifs par exemple.
- Largement extensible par l'ajout de modules complémentaires, il permet **l'acquisition de données**, notamment à l'aide de l'interface GPIB disponible sur de nombreux appareils, ou encore en pilotant des cartes d'entrées-sorties analogiques et digitales.

[1] Un « didacticiel » visant à présenter, *en français*, les bases du logiciel, en se concentrant sur les choses réellement utiles *dans le contexte de la préparation à l'agrégation*, constitue un part importante du TP « Outils Informatiques » de la série 0.

1 Objets et concepts fondamentaux

L'utilisation du logiciel conduit à manipuler des objets de différentes natures, que l'on peut classer en *données* d'une part, et en *objets d'interface graphique* d'autre part. Les données élémentaires manipulées par IGOR Pro peuvent être des **nombres** (entiers, réels, ou complexes) ou des **chaînes de caractères**. Toutefois, on utilise aussi des structures plus complexes, que nous allons décrire.

1.1 Les « Waves »

Le type de donnée fondamental dans le logiciel est appelé **wave** (en abréviation de « waveform »), et que l'on pourrait traduire en français par « signal ». Pour IGOR Pro, un « signal » est une liste de nombres (ou éventuellement de chaînes de caractères), qui peut être manipulée comme un vecteur. À ce titre, on peut lui attribuer une unité, des volts par exemple, nécessairement la même pour tous les nombres de la liste. Notons qu'IGOR Pro gère automatiquement les (sous)multiples k, M, G, T ou m, μ, n, p, f etc...

Les nombres de cette liste sont invariablement numérotés de 0 à N-1, où N est le nombre de points du « signal » considéré. En outre, chaque wave possède une échelle « horizontale », déduite des numéros de points par une transformation affine, correspondant à la valeur initiale (i.e. du premier point) et à l'écart entre les points successifs de la liste. Cette échelle possède elle-même une dimension. Une wave *semble* ainsi constitué de deux vecteurs, X (l'échelle) et Y (les données), à ceci près que les points de X, calculés, sont nécessairement des réels, équidistants et classés de façon monotone, alors que ceux de Y sont quelconques. Signalons d'ailleurs qu'ils peuvent même prendre les valeurs spéciales « *NaN* » (Not-a-Number), *Inf*, ou *-Inf*.

L'exemple typique est celui d'une courbe lue sur un oscilloscope numérique doté d'une interface GPIB ou RS232 (*cf.* § 3.2 pour la définition de ces termes) : les valeurs Y (en volts) sont lues l'une après l'autre et résultent d'autant de mesures, tandis que les valeurs en X (en secondes), sont les instants d'échantillonnage, simplement déduits de la base de temps. On observera sur la figure 1 que la table fait bien apparaître trois colonnes : les numéros de points, les abscisses X (visiblement équidistantes), et les ordonnées Y.

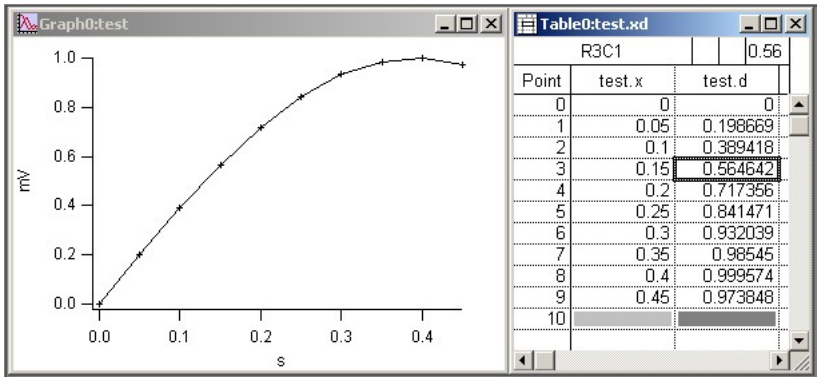


FIG. 1: Visualisation dans IGOR Pro d'une wave de 10 points,nommée « `test` », à gauche par un graphique, à droite par une table.

Le logiciel IGOR Pro permet aussi de manipuler et d'afficher des waves multidimensionnelles (de une à 4 dimensions), qui sont dotées d'autant d'échelles que de dimensions. En particulier, les

images numériques (« bitmaps ») sont commes des waves bidimensionnelles (en fait trois waves bidimensionnelles pour une image en couleurs RVB ou une wave 3D, une seule wave 2Dpour des niveaux de gris), et on pourra donc définir des échelles « spatiales » pour les deux directions du plan X et Y. L'ensemble des outils de traitement des images pourra être activé en sélectionnant `Analysis` ▷ `Packages` ▷ `Image Processing`.

Il faut souligner que les « waves », identifiées chacune par son nom, existent dans la mémoire de l'ordinateur de façon totalement indépendante de leur représentation sur un graphique ou dans une table, contrairement au données d'un tableur ! Une wave qui n'est pas utilisée ni affichée pourra être supprimée (`Data` ▷ `Kill Waves`) afin de libérer de la mémoire.

1.2 Graphiques et tables

Il s'agit des outils d'interface graphique de base, et donc les plus utilisés. On trouve un exemple de chacun des deux sur la figure 1.

Un **graphique** permet d'afficher un nombre illimité de waves, avec éventuellement un type d'affichage (format) et un système d'axes différents pour chacune. On peut ouvrir autant de graphiques que l'on souhaite, chacun dans une fenêtre séparée, et ils peuvent sans restriction utiliser les mêmes waves. Ils sont liés de façon *dynamique* aux waves : toute modification d'une wave entraîne la mise à jour instantanée du ou des graphique(s). On dispose pour chaque graphique de deux curseurs (A et B) qui peuvent être soit « liés » à une wave, soit « libres ». Les fenêtres graphiques peuvent enfin être dotées d'outils de contrôle, à des fins diverses de dépouillement ou d'acquisition de données. De nombreux exemples seront données dans la section 2.

Les **tables** permettent quant à elles d'afficher sous forme *numérique* autant de waves que l'on veut, dans l'ordre que l'on choisit, avec ou sans les valeurs de X. Elles sont un moyen commode de saisir ou modifier des valeurs, d'ajouter ou supprimer des points, ainsi que de renommer des waves ou encore de les supprimer ! À nouveau, les waves sont liées de façon dynamique aux tables qui les représentent.

1.3 La ligne de commande

Une particularité et une grande force d'IGOR Pro réside dans une petite fenêtre, appelée « fenêtre de commande » située en bas de l'écran et nommée comme le fichier en cours. On y distingue deux zones : en bas la « ligne de commande », et au dessus l'« historique des commandes ». Chaque opération (ou presque) exécutée à l'aide des menus déroulants ou des diverses boîtes de dialogue fait apparaître la commande correspondante dans la ligne de commande, puis dans l'historique une fois la commande achevée.

Les avantages de ce mode de fonctionnement sont les suivants :

- La présence de l'historique permet de garder une trace de toutes les opérations réalisées, ce qui est très précieux pour retrouver plus tard les modifications apportées au données initiales ou les valeurs de différents paramètres ;
- en cas de commandes répétitives (ou erronées !), il est très facile de rappeler la ligne correspondante, de l'éditer éventuellement, et de l'exécuter sans repasser par les menus ;
- certaines opérations rendant un grand nombre de données (comme un ajustement de courbes par exemple) peuvent écrire leur résultats dans l'historique, où ils sont disponibles pour utilisation ultérieure ;

- l'usage des menus permet d'apprendre progressivement les commandes, ce qui est précieux pour accélérer certaines tâches ou aborder la programmation ;
- la ligne de commande permet de réaliser un grand nombre d'opérations qu'il serait plus ou moins pénible de réaliser avec des boîtes de dialogue, ou avec une interface de type tableur, en particulier l'arithmétique sur les waves.

Le dernier point de cette liste justifie à lui seul l'existence de la ligne de commande. Ainsi par exemple, si `sig0`, `sig1`, et `sig2` sont des signaux de type réel et de même longueur, on pourra exécuter :

1. `Print 16*pi` qui écrira 50.2655 dans l'historique ;
2. `Variable d=sqrt(2)` qui crée une variable nommée `d` et lui affecte la valeur numérique de $\sqrt{2}$;
3. `sig0=x`, où la variable prédéfinie `x` représente l'abscisse de chaque point de `sig0`, compte tenu de l'échelle préalablement donnée à la wave `sig0`, ce qui revient à faire $Y = X$ pour chaque point ;
4. `sig1=Sin(sig0)/d` qui écrira dans chaque point de `sig1` le sinus du point correspondant de `sig0`, divisé par le nombre `d` ;
5. `sig2=BessJ(2,x)` qui tabulera dans `sig2` la fonction de Bessel J_2(`x`) (`x` défini par l'échelle de `sig2`) ;
6. `sig1=sig2[2*p]`, où la variable prédéfinie `p` représente le numéro de chaque point de `sig1`, qui copie dans la première moitié de `sig1` une sur deux des valeurs de `sig2` ;
7. `Edit sig0, sig2.id as "matable"` qui créera une nouvelle table, nommée "`matable`", et affichant les valeurs Y de `sig0`, ainsi que les valeurs X et Y de `sig2`.

D'une façon plus générale, on écrira dans la ligne de commande soit une *affectation* (cas 2–6) soit une *opération*, comme `Edit`, `Print`, etc (cas 1 et 7). Dans un cas comme dans l'autre, les expressions utilisées peuvent impliquer des fonctions prédéfinies dans IGOR Pro, ou définies par l'utilisateur. Dans tous les cas, les variables ou waves impliqués doivent soit exister, soit être explicitent créées par l'opération réalisée.

1.4 Les autres objets

On peut être amené à utiliser d'autres types d'objets que les graphiques et tables évoqués plus haut, que nous allons énumérer en indiquant sommairement leur utilité :

- les *notebooks* sont de simples fichiers textes, qui peuvent servir à documenter les données et leur traitement, et constituer une sorte de « cahier de manip » électronique ;
- les *control panels*, qui sont des fenêtres purement dédiées à l'interface utilisateur, en général avec un programme qui y lit et/ou écrit différents paramètres ;
- les *page layouts* qui, comme leur nom l'indique, permettent de mettre en page un certain nombre de graphiques et de tables, éventuellement assortis de commentaires, en vue de l'impression ;
- les *fenêtres d'aide*, qui sont très proches des notebooks, si ce n'est qu'on ne peut les modifier (`;-)`), et qu'ils ont généralement une structure hypertexte ;
- **la** *fenêtre de procédures* nommée Procedure, généralement cachée, qui contient :
  - les fonctions de « fit » définies par l'utilisateur,
  - les procédures de ré-création des graphiques, tables, panels et layouts que IGOR Pro propose de créer automatiquement lorsqu'on détruit l'un de ces objets,

  - toute macro écrite par l'utilisateur, en s'inspirant de l'historique...
- **les** *fenêtres de procédure*, qui contiennent des fonctions définies par l'éditeur ou par l'utilisateur, dont en particulier celles nécessaires à l'acquisition de données.

1.5 Les fichiers

Pour finir, il faut bien parler des fichier utilisés par IGOR Pro. L'ensemble des données, et les différents objets graphiques, tables, panels, notebooks, procédures, etc, d'une même session constituent pour IGOR Pro une « expérience » (experiment). Cet ensemble peut être stocké comme un tout dans un unique fichier .pxp, ou dans un répertoire où chaque wave et chaque objet sont dans un fichier séparé (format .uxp).

Pour des raisons évidentes d'organisation, il faut absolument utiliser le format compacté .pxp. Notons que l'on peut toujours, si nécessaire, exporter des données ou des graphiques avec les commandes idoines. Vous pouvez aussi être amenés à utiliser des fichiers modèles (templates) d'extension .pxt, qui contiennent déjà un certain nombre d'objets, de procédures, voire de données. Bien sûr, vous ne devrez en aucun cas enregistrer votre travail dans ce format !

Pour l'acquisition et le traitement des données, on sera amené à utiliser (de façon aussi transparente que possible !) des *fichiers de procédure*, d'extension .ipf, qui sont chargés par IGOR Pro en fonction des besoins, et qui nécessitent d'être compilés (menu Macro ▷ Compile) pour que les fonctions qu'ils définissent soient disponibles.

1.6 Programmation pour IGOR Pro

Il n'est bien sûr pas question, dans le contexte de la préparation à l'agrégation, d'apprendre à programmer pour IGOR Pro. Néanmoins, il est indispensable d'avoir en tête quelques idées générales sur le sujet pour ne pas perdre (trop) de temps si quelque chose ne « marche pas »...

Les « fonctions » (au sens général du terme) prédéfinies dans IGOR Pro sont de deux types :

- les *fonctions* proprement dites, qui s'écrivent avec des parenthèses, et prennent un nombre défini (mais éventuellement nul) d'arguments, et retournent un nombre ou une chaîne de caractères (mais en aucun cas une wave)
- les *opérations*, qui s'écrivent sans parenthèses, prennent un nombre variable d'arguments, et réalisent une action donnée sur leurs arguments, mais ne retournent pas de valeur.

Pour étendre les fonctionnalités d'IGOR Pro, c'est à dire lui faire exécuter des tâches ou des calculs qui ne sont pas assuré(e)s par les fonctions et opérations prédéfinies, il devient nécessaire de programmer. Cela prend deux formes nettement distinctes :

- On écrit en C++ ou en Fortran une DLL selon certaines règles pour qu'elle puisse s'interfacer avec IGOR Pro, et on crée ainsi une « extension » appelée « eXternal OPeration », ou XOP en abrégé. À condition d'être chargée par IGOR Pro au démarrage, cette extension, contenue dans un fichier binaire de type .xop, fournit de nouvelles opérations et fonctions.
- On reste dans IGOR Pro, et on écrit des « procédures » à l'aide des fonctions ou opérations fournies par IGOR Pro et ses extensions, dans un langage de script propre au logiciel, mais qui ressemble à la fois au C et au basic.

C'est la seconde méthode, plus légère, que nous avons employée pour développer les programmes d'acquisition de données que nous mettons à votre disposition. Ces programmes reposent fortement sur l'utilisation de XOP dédiées à l'interfaçage avec les appareils de mesure[2].

[2]Ces XOP sont nommément : `GetInputState.xop, VDT.xop, NIGPIB.xop, NIDAQ.xop, SoundInput.xop, VFWGrabber.xop`

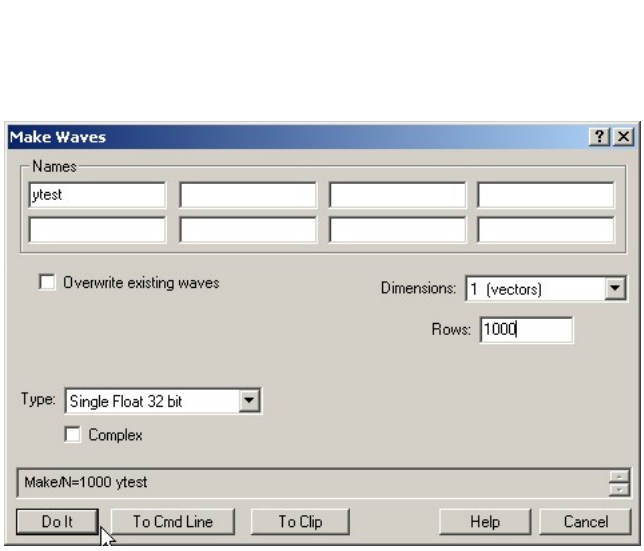
Les « procédures » écrites dans IGOR Pro sont elles-mêmes de divers types :
- Les **macros** et les **procédures** sont une successions de commandes telles qu'entrées sur la ligne de commande, avec des possibilités réduites de tests et de boucles. Les macros ne diffèrent des procédures que par leur déclaration, et apparaissent dans le menu Macros. Les macros et procédures ne sont pas compilées mais seulement *interprétées* lors de leur exécution.
- Les procédures de définition de graphiques, tables, panneaux, mises en pages, etc... sont des macros particulières, dont la déclaration définit le type d'objet créé ; elles apparaissent dans les menus Windows ▷ Graph Macros ou Windows ▷ Table Macros, etc...
- Les **fonctions**, déclarées comme `Function Nom_de_fonction`, sont des procédures de structure plus élaborées, pouvant utiliser des variables locales, et qui sont réellement compilées. Elles renvoient éventuellement une valeur de type nombre ou chaîne de caractère. Comme les procédures, elles ne figurent *a priori* dans aucun menu, mais peuvent être appelées depuis la ligne de commande. Nous avons préféré ce type chaque fois que cela était possible.

2 Un exemple simple

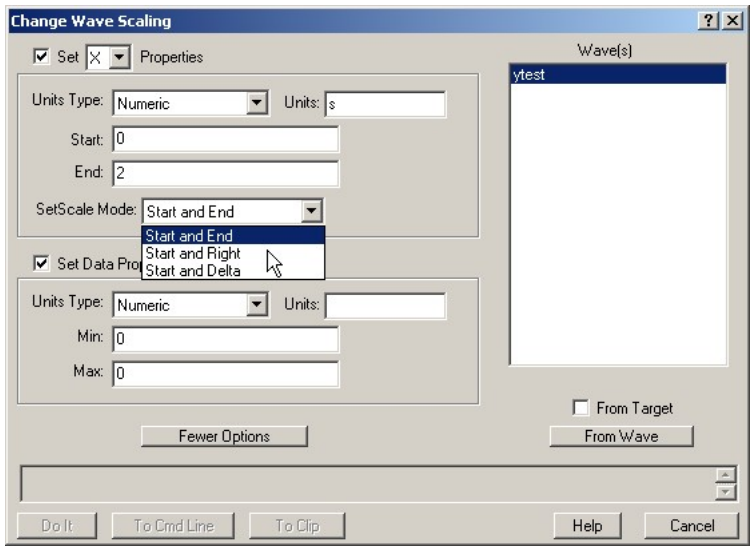
Lorsqu'on lance Igor, on voit défiler dans la ligne d'état (en bas de l'écran) les noms des XOP qu'il charge au démarrage). On obtient une table vide, que l'on peut fermer pour l'instant, et la fenêtre de commande nommée Untitled. Si l'on fait File ▷ Save Experiment As on peut enregistrer l'expérience avec un nom convenable (disons test), **en prenant garde à enregistrer dans le dossier Mes Documents et non dans le dossier Igor Pro Folder !** La fenêtre de commande est alors nommée test.

Manipulation et visualisation de données à l'aide des menus

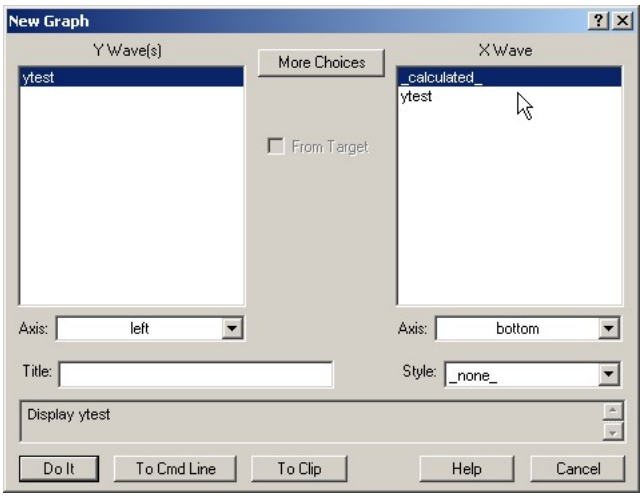
1. Sélectionner Data ▷ Make Waves et remplir la boîte de dialogue comme Fig 2-a
2. Sélectionner Data ▷ Change Waves Scaling et remplir la boîte de dialogue comme Fig 2-b, de façon à ce que X varie de 0 à 2 (prendre garde à bien sélectionner le mode Start and End) ;
3. Saisir dans la ligne de commande : `ytest=exp(-2*x)*sin(30*x)` et valider ;
4. Sélectionner Windows ▷ New Graph et valider en prenant soin de choisir `_calculated_` pour les valeurs de X (cf comme Fig 2-c) ;
5. Sélectionner Data ▷ Duplicate Waves, s'assurer que le modèle est bien `ytest`, et saisir le nom `xtest` puis valider ;
6. Pointer dans l'historique la ligne `ytest=...` et valider ; la formule apparaît dans la ligne de commande : l'éditer pour obtenir `xtest=exp(-2*x)*cos(30*x)` et valider ;
7. Sélectionner Graph ▷ Append Traces... et dans la boîte de dialogue, sélectionner `xtest` pour Y et `_calculated_` pour X en conservant les axes par défaut ;
8. Double-cliquer dans le graphique pour obtenir la boîte de format des courbes : la remplir comme Fig 2-d ;
9. les courbes obtenues peuvent être vues comme la position et la vitesse d'un oscillateur amorti ; pour obtenir un portrait de phase, il suffit de tracer `ytest` en fonction de `xtest` : sélectionner Windows ▷ New Graph, et choisir les waves Y et X en conséquence ;



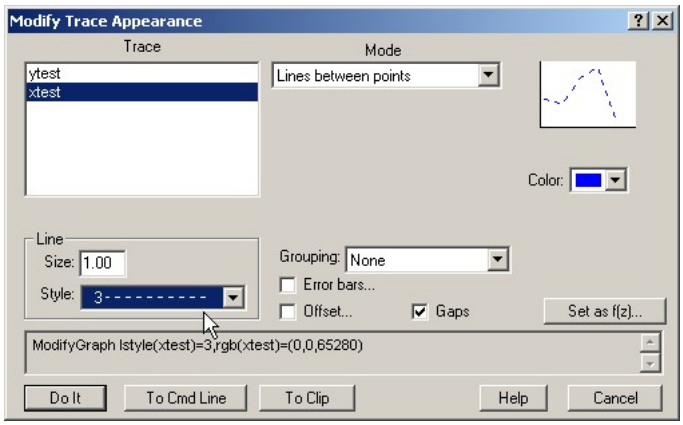
(a)



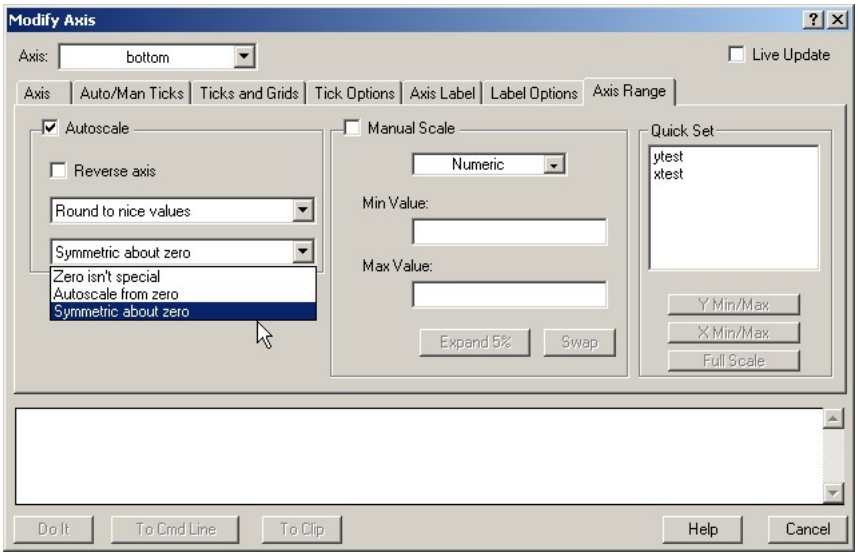
(b)



(c)



(d)



(e)

FIG. 2: Boîtes de dialogue de l'exemple

10. pour obtenir des axes plus appropriés, double-cliquer sur l'un des axes, pointer l'onglet Axis range et sélectionner les options `Round...` et `Symetric...` comme sur la Fig 2-e ; répéter la même procédure pour l'autre axe.

L'écran d'IGOR Pro doit alors avoir une apparence proche de la figure 3.

Enregistrer l'expérience. Fermer IGOR Pro, puis le réouvrir. Sélectionner File ▷ Recent Experiments, et choisir le fichier en question. Vérifier que tous les éléments, graphiques, waves, et autres se retrouvent dans l'état initial.

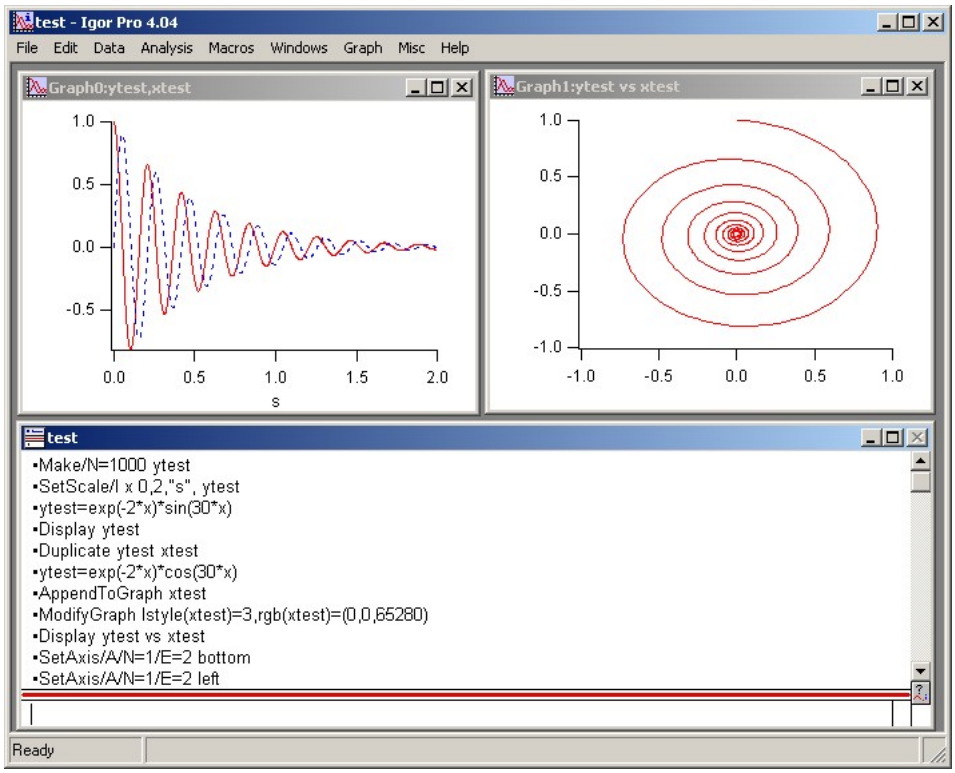


FIG. 3: Résultat de l'exemple

B – ACQUISITION DE DONNÉES

3 Généralités sur l'acquisition de données

L'acquisition de données par ordinateur, permet de rendre *certaines* mesures plus rapides, plus systématiques, et plus aisément exploitables. En plus des capteurs et appareils de mesure usuels, elle nécessite plusieurs ingrédients essentiels[3] :

1. un convertisseur analogique-numérique ;
2. une interface, c'est à dire un moyen de communication entre les appareils et l'ordinateur ;
3. un logiciel pour contrôler les uns et les autres et enregistrer les données ; parfois, le logiciel « pilote » en outre un paramètre de l'expérience.

3.1 Les convertisseurs

Les convertisseurs analogiques-numériques (et numériques analogiques) figurent au programme de l'agrégation, et sont donc hors de notre propos. Il suffit de savoir que l'on dispose – entre autres – des matériels suivants :

- des multimètres numériques (spécialement HP34401A) et de GBF numériques (spécialement HP33120A) ;
- des oscilloscopes numériques (spécialement TDS1012/3014 et HP54600) ;
- des cartes multifonctions (AD, DA, compteurs, etc...) directement intégrées dans l'ordinateur (spécialement NIDAQ PCI-6014) ;
- des caméras vidéo numériques au format « Webcam ».

3.2 L'interfaçage

Il existe de nombreux moyens de communications disponibles entre l'ordinateur et ses « périphériques », dont les appareils de mesure font partie. Chacun de ces moyens repose sur une architecture matérielle bien précise, généralement appelée « bus », et à une méthode bien précise d'échange, appelée « protocole », qui est une sorte de langage minimal commun entre les deux « interlocuteurs ».

Le terme de « bus » désigne de façon plus précise un ensemble de de lignes conductrices dont certaines sont destinées à coder les données binaires, et d'autres à indiquer quels sont l'émetteur et le destinataire des données (de façon générique les « adresses » de ceux-ci), ainsi que des signaux de dialogue, donnant des informations du type « données prêtes », « données lues », « envoyez la suite », « fin de communication », etc..., communément appelées « handshaking ». Sur un bus, les différents périphériques sont *en général* connectés en parallèle, mais ce n'est pas toujours le cas, et certains ne permettent de communiquer qu'avec un nombre restreint de périphériques (de 1 à 16).

La liste de tous les bus et interfaces utilisés dans un ordinateur évolue en permanence ; voici la liste, délibérément simplifiée, de ceux qui peuvent srvir à l'interfaçage, en commençant par les bus parallèles :
- les bus IDE/ATA (pour Integrated Drive Electronics / AT-Attached) et SCSI (pour Small Computer Systems Interface) servant pour la communication avec des périphériques de stockage ;
- le bus d'extension PCI (pour Peripheral Component Interconnect) permettant la connexion de

[3] La liste qui suit doit être comprise de façon générique, et chacun de ses éléments pourrait donc être au pluriel.

nombreuses « cartes filles », qui complète ou remplace le vieux bus ISA (pour Industry Standard Architecture) ;
- le bus GPIB (pour General Purpose Interface Bus), ou IEEE488, qui est spécialement utilisé dans le domaine de l'instrumentation numérique ;
- le LPT (pour Local Printer Terminal) ou port parallèle, ou encore IEEE 1284, utilisé par les imprimantes ou d'autres périphériques ;
et pour les bus série :
- les ports série COMx, ou RS232 (pour Recommended Standard 232) ;
- le bus USB (pour Universal Serial Bus) ;
- le bus « FireWire® » ou « iLink® » ou IEEE 1394, bus série rapide, principalement utilisé par les périphériques de vidéo numérique haute résolution

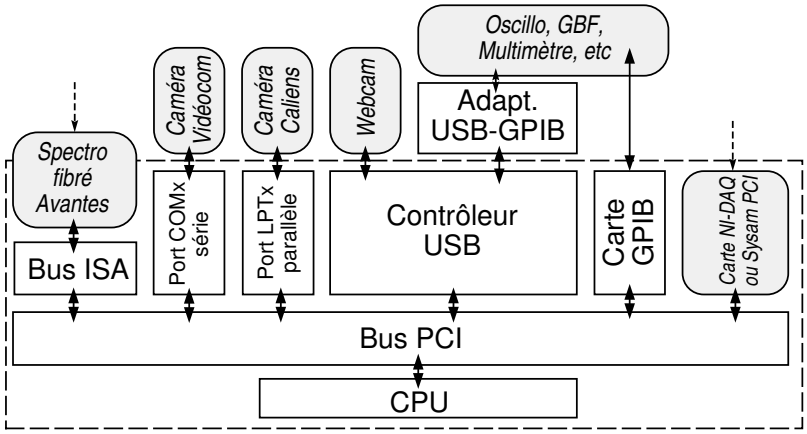


FIG. 4: *Schéma fonctionnel des différentes interfaces, et des appareils (en grisé) qui s'y connectent ; le rectangle pointillé délimite le boîtier de l'ordinateur.*

Dans les ordinateurs de type PC actuels, le lien fondamental entre l'ordinateur et les périphériques *externes* est le bus PCI. Certains appareils, et notamment les cartes AD-DA comme la carte NIDAQ PCI-6014, y sont directement connectes, dans l'un des « slots » disponibles sur la carte-mère. Les autres appareils utilisent des liaisons qui sont « cascadés » au moyen d'adaptateurs intermédiaires comme schématisé sur la figure 4. C'est le cas en particulier de l'interface « GPIB », qui est la plus commune sur les appareils pouvant fonctionner de façon autonome. Néanmoins, cela est totalement transparent pour l'utilisateur, qui doit simplement sélectionner le bon connecteur sur le boîtier de l'ordinateur.

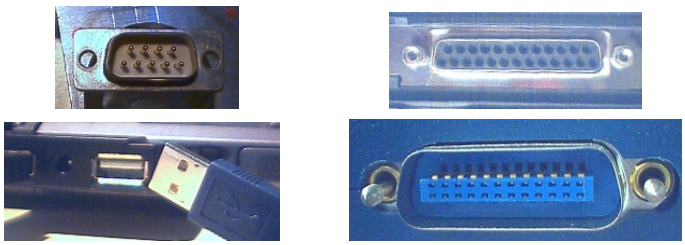


FIG. 5: *En haut, à gauche : port RS232, à droite : Port parallèle. En bas, à gauche : port et prise USB, à droite : port GPIB.*

Concrètement, en plus des cartes déjà connectées sur le bus PCI, on utilisera pour l'acquisi-

tion de données les interfaces suivantes, dont les connecteurs sont représentés à la figure 5 :
- la liaison « série » (ou `RS 232`), avec les ports (mono-instrument) `COM1` et `COM2` ;
- la liaison « parallèle », (ou `IEEE 1284`), avec le port `LPT` ou `Imprimante` ;
- la liaison `USB`, (liaison série multi-périphérique) ;
- enfin, et surtout, la liaison « GPIB ».

3.3 Le logiciel

Un certain nombre des appareils dotés d'une interface numérique et permettant l'acquisition de données informatisée utilisent un logiciel spécifique. Nous renvoyons en annexe une table donnant très succinctement la désignation de ces appareils, de l'interface utilisée, du logiciel correspondant, et de l'usage de ceux-ci. Pour le mode d'emploi, on se reportera aux notices correspondantes.

On donne ci-dessous une description sommaire des programmes d'acquisition de données que nous avons développés en utilisant les grandes possibilités d'IGOR Pro dans ce domaine. Ces programmes, conçus de façon à être relativement « génériques », se prêtent à de nombreux types d'acquisition de données et de traitement *dans un seul et même environnement*. Tous ces programmes sont accessibles via le menu dynamique Acquisition que nous avons ajouté à IGOR Pro (*cf.* figure 6). Pour leur utilisation, on se reportera aux notices correspondantes.

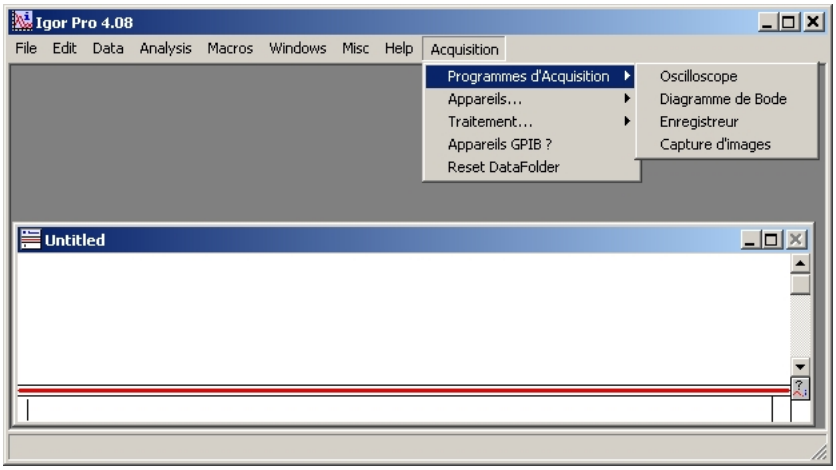


FIG. 6: *Menu* Acquisition

Le chargement d'un programme utilisant un ou plusieurs appareils peut toujours être fait, que le matériel soit présent ou non. Par contre, avant de commencer à communiquer avec un appareil, il est nécessaire *d'initialiser* la communication, en utilisant une procédure qui est disponible dans le sous-menu correspondant de Acquisition et/ou par un bouton « Init » accessible sur la fenêtre de contrôle de l'expérience.

Dans le cas – le plus fréquent – de la connexion GPIB, une fenêtre s'ouvre demandant de confirmer les adresses de la carte (conserver `"gpib0"`), et de l'appareil, sous la forme `"devX"`, où `X` est un nombre de 1 à 15, appelé « adresse GPIB » de l'appareil[4]. La valeur par défaut proposée correspond en général à l'adresse utilisée à Montrouge. Lorsque l'initialisation se passe bien, l'identité de l'appareil apparaît dans la fenêtre de commande. En cas d'échec, une fenêtre apparaît, invitant à vérifier l'adresse sur l'appareil, et sa bonne connexion. La commande Lister

[4]Cette adresse est fixée au niveau de l'appareil, à l'aide d'un menu approprié.

Appareils GPIB du menu Acquisition permet aussi d'obtenir une liste des appareils connectés sur l'interface GPIB, avec leur adresse[5].

Les programmes disponibles dans IGOR Pro sont les suivants :

1. « **Oscilloscope** » qui permet de relire, et d'exploiter dans IGOR Pro, les traces `ch1/2` et `Math` d'un oscilloscope numérique ;
2. « **Bode** » qui permet l'acquisition automatisée de la fonction de transfert de différents systèmes linéaires, en électronique en particulier, et le tracé et l'analyse des diagrammes de Bode correspondants ;
3. « **Enregistreur** », qui permet d'enregistrer sous forme numérique l'évolution (temporelle ou asynchrone) de différents signaux sur des appareils très divers : il s'agit en quelque sorte d'une super-« table traçante numérique » ;
4. « **sYnchroniX** », qui s'apparente à l'enregistreur, mais fonctionne beaucoup plus rapidement (jusqu'à 200 000 points par seconde), et/ou plus longtemps, grâce à l'usage de la carte PCI National Instruments (NIDAQ) ;
5. « **Capture d'image** » qui permet d'enregistrer des images fixes, à l'aide de la Webcam, puis de les traiter.
6. « **Dynamique** » qui permet d'importer des vidéos enregistrées grâce à la Webcam, et d'analyser le mouvement des mobiles enregistrés.

Remarques importantes

1 Nous avons choisi de ne pas *régler* les appareils par logiciel : avec ou sans ordinateur, il incombe aux utilisateurs de choisir les calibres, modes de déclenchement, et autres paramètres pertinents.

2 Tous les contrôles (ou presque) utilisés pour cl'acquisition possèdent une aide succincte qui apparaît dans la barre d'état (bas de la fenêtre) lorsque le pointeur de la souris les « survole ».

3 Les fenêtres principales sont des interfaces pour l'acquisition, et *ne doivent pas* servir au traitement, sous risque de compromettre les séquences d'acquisition ultérieures. Les waves qu'elles affichent sont le plus souvent inaccessibles à l'utilisateur, et un bouton permet d'en créer des copies pour traitement, dans des fenêtres appropriées.

Comment ça marche ?

IGOR Pro utilise pour étendre ses possibilités un type particulier de librairies appelées XOP (pour eXternal OPerations), ou extensions. Un certain nombre d'entre elles fournissent des interfaces vers les matériels d'acquisition de données, et nous avons plus particulièrement utilisé les suivantes :

- `NIGPIB.xop` permet la communication GPIB avec tous les appareils possédant cette interface, pourvu que l'on utilise un adaptateur de la marque National Instruments ;
- `NIDAQ.xop` permet de piloter les cartes d'acquisition du même fournisseur ;
- `VDT.xop` gère la communication série ;
- `VFWGrabber.xop` et `GetClipboardImage.xop` permettent la capture d'image ;

Dans un système convenablement configuré, ces extensions sont automatiquement chargées par IGOR Pro lors de son démarrage.

[5] Si l'échec persiste, il peut être nécessaire de fermer IGOR Pro (en enregistrant les données si l'on souhaite), puis de ré-ouvrir le programme.

ANNEXES

A Appareils fonctionnant avec un logiciel spécifique

Appareil	Logiciel	Usage	Interface	♯
Spectroscope	Avasoft 5	Analyse spectrale de la lumière (visible)	ISA/USB	1/1
Barrette CCD Caliens	Caliens*	Enregistrement de figures de diffraction	LPT	1
Carte AD-DA Sysam	Synchronie 2003 **	Digitalisation de signaux électriques	PCI	1
Carte AD-DA NIDAQ	IGOR Pro	*voir notice spécifique*	PCI	3
Webcam Philips	AmCAP***	Capture d'images fixes ou de vidéo	USB	2
Caméra Leybold	Vidéocom****	Analyse du mouvement d'un objet	COM	1

Paires matériel/logiciel pour l'acquisition de données informatisée.
* *: peut aussi être exploitée dans IGOR Pro, en lisant le signal vidéo sur un oscillo numérique ;*
** *: des fonctions très similaires sont disponibles dans le programme IGOR Pro « sYnchroniX » ;*
*** *: peut aussi être exploitée dans IGOR Pro : voir notice « Capture d'image » ;*
**** *: peut généralement être remplacée par la webcam, avec analyse du film dans IGOR Pro.*

Contexte	Menu	Nom	Commande	Raccourci	Fonction
Général	Help	Help Browser		F1	Aide générale
	File	New Experiment		Ctrl-N	Nouvelle expérience[a]
	File	Save Experiment		Ctrl-S	Enregistre données et fenêtres
	File	Load Waves		Ctrl-L	Charge des données depuis un fichier[b]
	File	Make New Waves	Make		Crée une (des) nouvelle(s) wave(s)
	Data	Duplicate Waves	Duplicate		Recopie une wave
	Data	Change wave scaling	SetScale		Définit les échelles
	Data	Browse Waves			Liste des waves avec leurs propriétés
	Data	Compute New Waves*	MakeScale	Ctrl-1	Crée des waves et calcule les données*
	Windows	Command Win.		Ctrl-J	Fenêtre de commande au premier plan
	Windows	Procedure Win.		Ctrl-M	Fenêtre de procedure au premier plan
	Windows	New Graph	Display		Crée une nouvelle table
	Windows	New Table	Edit		Crée un nouveau graphique
	Windows	New ▶			Vue d'image, de wave complexe* etc...
	Windows	Graphs ▶			Remet un graphique au premier plan
	Windows	Tables ▶			Remet une table au premier plan
	Windows	Close		Ctrl-W	Ferme la fenêtre du premier plan
				Ctrl-Alt-W	idem sans dialogue
Ligne de commande				Ctrl-F1	Insère modèle de fonction[c]
				Ctrl-Alt-F1	Ouvre l'aide sur la fonction[c]
				Ctrl-K	Efface la ligne de commande
				Ctrl-Alt-Clic	Réexecute une ligne de l'historique
				Clic + Enter	Rappelle et édite une commande
Graph ou Table	Graph/Table	Append Traces/Columns	Append		Ajoute des waves
	Graph/Table	Remove	Remove		Retire des waves
	Graph	Show Info	ShowInfo	Ctrl-I	Affiche le curseurs
	Graph	Autoscale Axes	SetAxis/A	Ctrl-A	Axes en échelle automatique
	Graph	Modify Axis	ModifyGraph	Dbl-Clic	Modifie la présentation des axes
	Graph	Modify Trace(s)	ModifyGraph	Dbl-Clic/R-Click	Modifie la présentation des waves
Analyse de données	Analysis	Curve Fitting	CurveFit / FuncFit		Réalise un ajustement
	Analysis	Wave Stats	WaveStats		Moy., Ecart, RMS, Max, Min ...
	Analysis	Fourier Transform	FFT / IFFT		Transformée de Fourier discrète[d]
	Analysis	Sort	Sort		Trie les waves "destination"
	Analysis	Integrate/Differentiate	*idem*		Intégration / Différentiation discrète[d]

$*$:obtenu par une procédure. a : réinitialise tout. b : interactif, nombreuses options. c : le nom de fonction doit être complet et exact. d : Ecrase la wave source avec le résultat.